CIT360

Portfolio #8 (Week8)

03/12/2016

Jinseong Jeong

**Java Collections – List**

The list interface extends collection and declares the behavior of a collection that stores a sequence of elements.

Example: Interface has been implemented in various classes like ArrayList or linkedList, etc. Following is the example explain few methods from various class implementation of the above collection method:

Example code:

<https://github.com/jinseongv/homework/blob/master/CollectionList.java>

This would produce the following result:

ArrayList Elements

[Zara, Mahnaz, Ayan]

LinkedList Elements

[Zara, Mahnaz, Ayan]

**Java Collection - Set**

The Set interface contains only methods inherited from Collection and adds the restriction that duplicate elements are prohibited. Set also adds a stronger contract on the behavior of the equals and hash Code operations, allowing Set instances to be compared meaningfully even if their implementation types differ.

Example: Set has its implementation in various classes like HashSet, TreeSet, LinkedHashSet.

Example code:

<https://github.com/jinseongv/homework/blob/master/CollectionSet.java>

This would produce the following result:

[amrood]$ java SetDemo

[34, 30, 60, 10, 22]

The sorted list is:

[10, 22, 30, 34, 60]

The First element of the set is: 10

The last element of the set is: 60

**Java Collections – Tree types**

TreeSet provides an implementation of the Set interface that uses a tree for storage. Objects are stored in sorted, ascending order. Access and retrieval times are quite fast, which makes TreeSet an excellent choice when storing large amounts of sorted information that must be found quickly.

Example: The following program illustrates several of the methods supported by this collection:

Example code:

<https://github.com/jinseongv/homework/blob/master/CollectionTree.java>

This would produce the following result:

[A, B, C, D, E, F]